```sql
--DAY 03

-- HAVING: 그룹에 대한 조건
/*실선, 직선   : 식별, 비식별 관계*/


SELECT*
FROMEMPLOYEE;

-- GROUP BY: 하위 데이터 그룹
-- GROUP BY COLUMN_NAME | EXPR
-- 부서별 평균 급여를 확인하고 싶다면?

SELECTDEPT_ID, AVG(SALARY) /*그룹함수는 GROUP BY와 함께 사용되는 경우가
FROMEMPLOYEE
GROUP BY DEPT_ID;

-- 성별에 따른 평균급여를 확인하고 싶다면?
-- GROUP BY는 별칭, 인덱스 사용이 불가

SELECT CASE SUBSTR(EMP_NO,8,1)
    WHEN '1' THEN '남자'
    WHEN '2' THEN '여자'
  END AS "성별",
  ROUND(AVG(SALARY),-4) "평균급여"
FROM EMPLOYEE
GROUP BYCASE SUBSTR(EMP_NO,8,1)
    WHEN '1' THEN '남자'
    WHEN '2' THEN '여자'
  END

/*가장 높은 부서의 급여 평균*/
/*오류 구문*/
SELECTDEPT_ID, MAX(AVG(SALARY))
FROMEMPLOYEE
GROUP BY DEPT_ID;

/*부서별 급여 총합이 900 이상인 부서 필터링*/
/*HAVING: 그룹에 대한 조건*/
/*WHERE 절에서는 그룹에 대한 조건 사용 불가*/
SELECTDEPT_ID, SUM(SALARY)
FROMEMPLOYEE
GROUP BY DEPT_ID
HAVINGSUM(SALARY) >= 9000000;  /*그룹에 대한 SUM*/

-- JOIN 문 정리
-- JOIN: 두 개 이상의 테이블을 논리적으로 병합하여 하나의 결과를 출력하는 것
-- 기본 조건: 부모 테이블의 기본키(PK) 와 자식 테이블의 외래키(FK) 를 비교
```

```sql

-- JOIN 문법 (ON 구문 | USING 구문)
-- JOIN 테이블명 ON 조건
-- JOIN 테이블명 USING (공통컬럼명)

-- JOIN 종류
-- EQUALS JOIN (= INNER JOIN)
-- NON-EQUALS JOIN : 업무적 연관관계 X


-- ORACLE 전용 JOIN 구문

SELECT*
FROMEMPLOYEE;

SELECTCOUNT(*)
FROMDEPARTMENT;

SELECTEMP_NAME,
  DEPT_NAME,
  JOB_TITLE
FROM  EMPLOYEE E, DEPARTMENT D, JOB J  /*테이블 병합*//*테이블에
WHERE E.DEPT_ID = D. DEPT_ID AND J.JOB_ID = E.JOB_ID;

-- ANCI 표준 JOIN
/*ON 과 USING은 WHERE을 의미
부모의 기본키와 자식의 외래키가 같으면 USING, 다르면 ON 사용
USING + COLUMN명
ON  + 조건
*/

SELECTEMP_NAME,
  DEPT_NAME,
  JOB_TITLE,
  LOC_DESCRIBE
FROMEMPLOYEE E
JOINDEPARTMENT D USING(DEPT_ID)
JOINJOB J USING (JOB_ID)
JOINLOCATION L ON(L.LOCATION_ID = D.LOC_ID);


SELECTEMP_NAME,
  DEPT_NAME
FROMEMPLOYEE E
FULL JOIN DEPARTMENT D USING(DEPT_ID);  /*OUTER JOIN*/


```

```sql
-- NON-EQUALS JOIN : 업무적 연관관계 X

SELECT*
FROMSAL_GRADE;

SELECTEMP_NAME,
  SALARY,
  SLEVEL
FROMEMPLOYEE E
JOINSAL_GRADE S ON (E.SALARY BETWEEN S.LOWEST AND S.HIGHEST)
ORDER BY 3;


/* SELF RECURSIVE RELATIONSHIP */

SELECTE.EMP_NAME, M.EMP_NAME, S.EMP_NAME
FROM  EMPLOYEE E
LEFT JOINEMPLOYEE M ON (E.MGR_ID = M.EMP_ID)
LEFT JOINEMPLOYEE S ON (M.MGR_ID = S.EMP_ID);

/*직급이 대리이고 지역이 아시아로 시작하는 사원의 이름과 직급, 부서이름을 조회하라.*/


SELECT*
FROM  EMPLOYEE;

SELECT*
FROM  JOB;

SELECT*
FROM  LOCATION;

SELECT*
FROM  DEPARTMENT;


SELECTEMP_NAME,
  JOB_TITLE,
  DEPT_NAME,
  LOC_DESCRIBE
FROM  EMPLOYEE E
JOINJOB J USING(JOB_ID)
JOINDEPARTMENT D USING(DEPT_ID)
JOINLOCATION L ON (L.LOCATION_ID = D.LOC_ID)
WHERE J.JOB_TITLE = '대리'
  AND L.LOC_DESCRIBE LIKE '아시아%';

/*ADDITIONAL SELECT-OPTION*/
```

```sql

/*1. 학생이름과 주소지를 표시하시오. 단, 출력 헤더는 "학생 이름", "주소지"로 하고,
정렬은 이름으로 오름차순 표시하도록 한다.(pg. 13) */

SELECT STUDENT_NAME AS "학생 이름",
  STUDENT_ADDRESS AS "주소지"
FROM  TB_STUDENT
ORDER BY STUDENT_NAME;


/*2. 휴학중인 학생들의 이름과 주민번호를 나이가 적은 순서로 화면에 출력하시오.*/

SELECT STUDENT_NAME, STUDENT_SSN
FROM TB_STUDENT
WHERE  ABSENCE_YN = 'Y'
ORDER BY STUDENT_SSN DESC;


/*3. 주소지가 강원도나 경기도인 학생들 중 1900 년대 학번을 가진 학생들의 이름과 학번,
주소를 이름의 오름차순으로 화면에 출력하시오. 단, 출력헤더에는 "학생이름","학번",
"거주지 주소" 가 출력되도록 한다.*/

SELECT ENTRANCE_DATE
FROM TB_STUDENT;

SELECT STUDENT_NO
FROM TB_STUDENT;


SELECT STUDENT_NAME AS "학생이름",
  STUDENT_NO AS "학번",
  STUDENT_ADDRESS AS "거주지 주소"
FROM TB_STUDENT
WHERE SUBSTR(STUDENT_NO,1,1) = '9'
  AND (STUDENT_ADDRESS LIKE '%강원도%'
  OR STUDENT_ADDRESS LIKE '경기도%')
ORDER BY 1 ASC;


/*4. 현재 법학과 교수 중 가장 나이가 많은 사람부터 이름을 확인할 수 있는 SQL 문장을
작성하시오. (법학과의 '학과코드'는 학과 테이블(TB_DEPARTMENT)을 조회해서 찾아
내도록 하자)*/

SELECT DEPARTMENT_NO,DEPARTMENT_NAME
FROM  TB_DEPARTMENT;

SELECT PROFESSOR_NAME,PROFESSOR_SSN
FROM  TB_PROFESSOR P
```

```sql
JOIN TB_DEPARTMENT D USING(DEPARTMENT_NO)
WHERE DEPARTMENT_NO = '005'
ORDER BY  PROFESSOR_SSN;


/*5. 2004 년 2 학기에 'C3118100' 과목을 수강한 학생들의 학점을 조회하려고 한다.
높은 학생부터 표시하고, 학점이 같으면 학번이 낮은 학생부터 표시하는 구문을 작성해보시오.
작성해보시오.*/

SELECT  CLASS_NO
FROMTB_CLASS;

SELECT*
FROMTB_GRADE;

SELECT STUDENT_NO, TO_CHAR(POINT,9.99)
FROM TB_STUDENT S
JOIN TB_GRADE G USING (STUDENT_NO)
WHERE TERM_NO = '200402' AND CLASS_NO = 'C3118100'
ORDER BYPOINT DESC;


/*6. 학생 번호, 학생 이름, 학과 이름을 학생 이름으로 오름차순 정렬하여 출력하는 SQL
문을 작성하시오.*/


SELECT STUDENT_NO, STUDENT_NAME, DEPARTMENT_NAME
FROM TB_STUDENT S
JOIN TB_DEPARTMENT D USING (DEPARTMENT_NO)
ORDER BYSUBSTR(STUDENT_NAME,2,2);


/*7. 춘 기술대학교의 과목 이름과 과목의 학과 이름을 출력하는 SQL 문장을 작성하시오.*/

SELECTCLASS_NAME,DEPARTMENT_NAME
FROMTB_CLASS C
JOINTB_DEPARTMENT D USING (DEPARTMENT_NO)
ORDER BY DEPARTMENT_NO, CLASS_TYPE DESC, CLASS_NAME ASC;


/*8. 과목별 교수 이름을 찾으려고 한다. 과목 이름과 교수 이름을 출력하는 SQL 문을
작성하시오.*/

SELECTCLASS_NAME, PROFESSOR_NAME
FROMTB_CLASS_PROFESSOR P
JOINTB_CLASS C USING(CLASS_NO)
JOINTB_PROFESSOR R USING(PROFESSOR_NO)
ORDER BYC.DEPARTMENT_NO, CLASS_TYPE DESC,PROFESSOR_NAME DESC,
CLASS_NAME ;
```

```
242
243
244
245 /*9. 8 번의 결과 중 '인문사회' 계열에 속한 과목의 교수 이름을 찾으려고 한다. 이에
246 해당하는 과목 이름과 교수 이름을 출력하는 SQL 문을 작성하시오.*/
247
248 SELECT CLASS_NAME, PROFESSOR_NAME
249 FROM TB_CLASS_PROFESSOR P
250 JOIN TB_CLASS C USING (CLASS_NO)
251 JOIN TB_PROFESSOR R USING (PROFESSOR_NO)
252 JOIN TB_DEPARTMENT D ON (D.DEPARTMENT_NO = R.DEPARTMENT_NO)
253 WHERE CATEGORY = '인문사회'
254 ORDER BY D.DEPARTMENT_NO;
255
256
257
258 /*10. '음악학과' 학생들의 평점을 구하려고 한다. 음악학과 학생들의 "학번", "학생
259 "전체 평점"을 출력하는 SQL 문장을 작성하시오. (단, 평점은 소수점 1 자리까지만
260 반올림하여 표시한다.)*/
261
262 SELECT STUDENT_NO AS "학번",
263  STUDENT_NAME AS "학생이름",
264  ROUND(AVG(POINT),1) AS "전체 평점"
265 FROM TB_STUDENT
266 JOIN TB_DEPARTMENT USING (DEPARTMENT_NO)
267 JOIN TB_GRADE USING (STUDENT_NO)
268 WHERE DEPARTMENT_NAME = '음악학과'
269 GROUP BY STUDENT_NO, STUDENT_NAME;
270
271
272 /*11. 학번이 A313047 인 학생이 학교에 나오고 있지 않다. 지도 교수에게 내용을
273 위한 학과 이름, 학생 이름과 지도 교수 이름이 필요하다. 이때 사용핛 SQL 문을
274 작성하시오. 단, 출력헤더는 ,학과이름', ,학생이름', ,지도교수이름'으로
275 출력되도록 한다.*/
276
277 SELECT DEPARTMENT_NAME AS "학과이름",
278  STUDENT_NAME AS "학생이름",
279  PROFESSOR_NAME AS "지도교수이름"
280 FROM TB_STUDENT S
281 JOIN TB_DEPARTMENT  USING (DEPARTMENT_NO)
282 JOIN TB_PROFESSOR P ON (S.COACH_PROFESSOR_NO = P.PROFESSOR_NO)
283 WHERE STUDENT_NO = 'A313047';
284
285 /*12. 2007 년도에 '인간관계론' 과목을 수강한 학생을 찾아 학생이름과 수강학기름
286 SQL 문장을 작성하시오.*/
287
288
289
```

```
290  TERM_NO
291 FROM TB_STUDENT
292 JOIN TB_GRADE USING (STUDENT_NO)
293 JOIN TB_CLASS USING (CLASS_NO)
294 WHERE CLASS_NAME = '인간관계론' AND SUBSTR(TERM_NO,1,4) = '2007'
295 ORDER BY 1;
296
297
298
299 /*13. 예체능 계열 과목 중 과목 담당교수를 한 명도 배정받지 못한 과목을 찾아 그 과목
300 이름과 학과 이름을 출력하는 SQL 문장을 작성하시오.*/
301
302 SELECT *
303 FROM TB_CLASS_PROFESSOR;
304
305 SELECT CLASS_NAME, DEPARTMENT_NAME
306 FROM TB_CLASS
307 LEFT JOIN TB_CLASS_PROFESSOR R USING (CLASS_NO)
308 JOIN TB_PROFESSOR P USING (DEPARTMENT_NO)
309 JOIN TB_DEPARTMENT D USING (DEPARTMENT_NO)
310 WHERE CATEGORY = '예체능' AND R.PROFESSOR_NO IS NULL;
311
312
313
314
315
```